Bryan Notes from 2nd long youtube video:

Spark runs 100x faster than MapReduce (in memory) and 10x faster on disc.

APIs: Python, Scala (Spark is written in Scala), and Java (MapReduce only uses Java)

Can run Spark on top of HDFS/YARN (if your data is already there), but can use standalone scheduler

RDD (Resilient Distributed Dataset) – this is the interface to your data. How is Spark looking at your data in HDFS? This is your RDD.   
  
RDD takes splits of your text file, assigns each split (partition/block) to a node, partitions can be stored in cache or disk (depending on your application – Spark can handle both). RDD views the partitions as seamless data. Input/Output options: HDFS, CSV, TXT, MD, Tab, JSON, Hive, HBase, JDBC, etc.

Two types or RDD operations:

1. Transformations – subset of original RDD, uses pointers to new RDD.
   1. Filters to weed out data you want/don’t want
   2. Map transformations perform calculation on values
   3. flatMap will break strings of words up into individual words
   4. Distinct will do exactly what you’d expect – weed out distinct values
   5. Sample will pull out sample data, ya think?
   6. Going to stop with notes now – just think logical SQL operators, RDD can perform these operations, but they are called “transformations”. Think union, subtract, Cartesian, etc.
   7. Pair RDD transforms (joins, keys, group by, etc.)
2. Actions – computations
   1. Count
   2. Reduce
   3. Collect
   4. Take (take 10 elements and use them)

Persistence – every time you call an action, Spark will re-compute all transformations. Persistence stores the actions in cache (or disk) so you aren’t re-doing the transformations over and over again.

Accumulators – similar to MapReduce counters. Tracks metrics for debugging. Allows all nodes to communicate with a single global variable (aka, the accumulator)

Broadcast variable – similar to MapReduce Distributed Cache. Sens read-only values to worker nodes, great for lookup tables (aka validation tables, lists of values) & dictionaries.

(count, mean, sum, min, max, variance, standard deviation)

Data Frames = “tables” in MySQL or a Data Frame in R. Allows for faster computations (in Python, a data frame is 5x faster than an RDD). Works for hive, hdfs, PostgreSQL, JSON, etc.

MLIB – machine learning, can do regression, classification, clustering, chi-square, correlation, summary stats, etc. Takes classic algorithms and parallelizes them across thousands of nodes (potentially).